EAM – For the new kid on the block

G'Day All,

Picking up from my previous topic aboutARA – For the new kid on the block, this document is just an overview of my understanding of what EAM is and how it works.

The objective of this document is to give people who are just starting out or even beginning to find their feet, a brief overview of EAM before they can get stuck into it and go all in(links provided). This is not intended for people who are well versed on this topic, so if that is you, please feel free to skip it as this might not interest you. However if you do want to stick around and point/correct any mistakes or offer advice/suggestions, please by all means do so. I am open to constructive criticism.

I understand there is a lot of content related to EAM in this site and some of the information covered herein might exist elsewhere in some shape or form however this is just meant to serve as a conduit for freshers, who might get a tad overwhelmed by all the information lying around. So I hope this document can give them a glimpse of what it is all about and then help them to venture out into the wild.

What is it all about?
EAM enables end-users to perform emergency activities outside the parameters of their standard role, but within a controlled and fully audit-able environment. The application assigns a temporary Firefighter ID that grants an end user(firefighter) broad yet regulated access, and logs every activity he/she performs using the temporary ID. This is usually done in emergency situations, where it is imperative for a user to execute certain tasks irrespective of SOD violations and transaction code clashes however all of his/her actions are monitored and recorded making the session completely visible and transparent.
Key challenges of EAM
1. Identification of Business Processes and creating dedicated Firefighter IDs/Roles pertinent to them. 2. Identification of the need for usage of Firefighter ID/Role 3. Identification of Firefighters, Firefighter Owners, Controllers, and Administrators. 4. Identification/Standardization of Reason Codes 5. Consistency of naming conventions for Firefighter ID/Roles and Reason Codes.

What is it all about?
6. Archival policy for the Firefighter Logs 7. EAM usage policy should be created to identify tasks which can be positively supported by EAM. 8. Last but not least, performance optimization.
Potential functional scenarios for EAM Access
Additional resources with additional roles Approaching month/financial year end and need additional resources to speed up certain activities. Additional resources are required but they don't have enough authorizations. This task can be easily automated by EAM and individual activity log would be generated for later review. **Developer access on production system** Developer access on production systems is one of the most critical scenarios, but at times it becomes necessary to allow developer access to fix certain bugs urgently. This is an ideal emergency scenario for assigning firefighter id to track each and every activity a developer or a group of developers perform. However developer access on production is never recommended but when you can't wait for a bug-fix to travel from a lengthy procedure (Dev-Qual-Prod) then EAM works as a mighty mitigation control. **Contract user access** To maintain track of contract users activities for a certain period of time. This can be achieved by assigning Firefighter IDs to contract users for access on the assigned system. This allows all their activities to be recorded for an extended review and hence management oversight is achieved. **Auditor Access** Most companies have strict audit procedures in place, which entails both internal and external auditors to conduct audits on a regular basis. Auditors can be granted temporary access through EAM. **** By no means is this list exhaustive however it should give you an indication of the***

Potential functional scenarios for EAM Access
potential reasons for EAM Access. **** Given the fact that EAM is a form of Mitigation (Please check the ARA document), It is used in scenarios where you have exhausted all other options!!***

Firefighter Users, Roles and Responsibilities	
Users/FFID/ FFROLE	**Roles & Responsibilities**
Firefighter ID	This is a unique user id, created with specific roles that allow the firefighter to perform the required tasks. So we can create multiple Firefighter id's with specific roles and assign them to the designated users (Firefighters) for a set period of time. • SU01: Create FFID • Roles: SAP_GRAC_SPM_FFID (This should be exactly the same in config settings as well. Shown further in the document)
Firefighter Role	This is a unique role, which gets assigned to the firefighter to perform the requited tasks. • PFCG/BRM: Create FFROLE. Ensure this role is enabled for firefighting in BRM.
Firefighter	These are the users who get assigned with the required Firefighter ID/Role. Firefighter users use Firefighter ID/Role to perform firefighting tasks. • SU01: Create FFighter or assign the role to an existing user • Role: SAP_GRAC_SUPER_USER_MGMT_USER (This role might need other additional authorizations. Please check the links provided)
Firefighter Administrator	This is the person who has got the ultimate authority over the firefighter program. He/she is responsible for assigning FF ID/roles to firefighters (if they choose to), Owners. They can generate reports, ensure reason codes are up to date etc. • SU01: Create FF ADMINISTRTOR or assign the role to an existing user • Roles: SAP_GRAC_SUPER_USER_MGMT_ADMIN, SAP_GRAC_BASE, SAP_GRAC_NWBC
Firefighter Owner	These are the ID/Role owners and are responsible for assigning FF ID/roles assigned to them by the administrator, to firefighters and controllers. They can also act as controllers however they should not be able to assign FF ID/roles to

Potential functional scenarios for EAM Access	
	themselves. They can only be one FF Owner per FF ID/role however one FF Owner can have multiple FF ID/roles. • SU01: Create FF Owner or assign the role to an existing user • Roles: SAP_GRAC_SUPER_USER_MGMT_OWNER, SAP_GRAC_BASE, SAP_GRAC_NWBC
Firefighter Controller	These are the people who monitor the actions of the firefighters. They can do this by viewing the log report and can even receive email notifications when a Firefighter logs in. • SU01: Create FF Controller or assign the role to an existing user • Roles: SAP_GRAC_SUPER_USER_MGMT_CNTLR, SAP_GRAC_BASE, SAP_GRAC_NWBC

**** All of the aforementioned roles can/needs to be customized. One can use a naming convention that suits their company requirements***

AC10 has the option of having either Centralized or Decentralized firefighting (more on this in the links provided at the end of the document).

Centralized

User has to go from plugin/backend system (R3PRD001) and log into a GRC System (GRCPRD001), execute GRAC_SPM(OR EAM) -> which will launch the EAM launchpad -> then access the system [R3PRD001 or something else (HCMPRD001), (CRMPRD001) etc] assigned to him/her by clicking the logon button-> perform FF tasks.

- This is a better option when in some companies, the user has to access multiple systems. So he/she can log into GRC system (GRC Box) and can start firefighter sessions by clicking on 'logon', which will take him/her to the assigned system.
- Firefighters can log on centrally as opposed to logging into multiple systems separately
- FFAdministrator, FFOwner, FFController, Firefighter and their respective roles have to be maintained in the GRC system
- FFID and its respective role has to be maintained only in the plug-in system

Decentralized

User has to stay on the BackEnd system (R3PRD001) execute /n/GRCPI/GRIA_EAM -> which will launch

Potential functional scenarios for EAM Access
the EAM launchpad -> then click the logon button to start a session in the very same system (R3PRD001) and perform FF tasks. You can enable DC-FF by parameter 1000: GRD(RFC Connector pointing to itself), 4015. • The most important advantage of DC firefighting is that you can continue using firefighter even when the GRC Box is down. • It's also more "user-friendly" since the firefighter doesn't have to log on to GRC Box in order to start the firefighting session, he/she only needs to execute a transaction in the plugin/backend system. • Firefighter and his/her respective role has to be maintained just in the plug-in system • FFID and its respective role has to be maintained only in the plug-in system • FFController and his/her respective role has to be maintained both in the plug-in/GRC system(to receive emails of logs) • FFAdministrator and FFOwner and their respective roles have to be maintained in the GRC system
ID Based vs Role Based
One of the key difference between assigning a Firefighter an FFID vs FFRole is added security. An FFID is built with a certain role in mind, which has predetermined tcodes assigned to it and this gets assigned to an end user (firefighter). So if this user wishes to commit fraud, he/she can execute certain tcodes from his/her user id and then the remaining from the FFID. This way the chances of him/her getting caught, is dependent on a thorough monitoring/analysis by the controller/auditors. Whereas if you build a specific firefighter role with the same tcodes, this role gets assigned to the end user not an FFID, so every transaction executed shows up against their user id, which makes his/her task of committing fraud a lot harder if not negligible. key differences are as follows:

ID Based vs Role Based	
ID Based	**Role Based**
Logs in using own user ID, accesses FFID from the GRC System and logs into the system assigned to them(ECC, SRM, CRM etc).	Logs into the plug-in system using own user ID, so everything gets logged against that one ID. Multiple users can use the FFROLE at once.
Only one user at a time can use a FFID.	Multiple users can use multiple FFRoles at once.
Firefighter need not exist in every system assigned to them due to central logon however they need to exist in the GRC system (This is only applicable for Centralised firefighting).	Firefighter has to exist in every system assigned to them – so multiple logons. (This is only applicable if the user needs to perform tasks in other systems).
Knows exactly when FFID is being used as he/she has to login so has a psychological effect (good thing).	Hard to differentiate between FF tasks and normal tasks as there is no login required. So easy to slip up.
Better tracking of FF tasks – Specific log reports with Reason Codes. Bonus point from Auditors!	Time consuming to track FF tasks – No Specific log reports. No Reason Codes.
Two logins so potential to commit fraud. (1 action using own UserID and 1 action using FFID).	Only one login, so everything gets logged against one id(own user id). Harder to commit fraud.
Could be hard to track and find out when a fraud has been committed so can be a problem with auditors. When two logins used	Easy to track as only one login is used however a thorough analysis is required to differentiate ff tasks from normal tasks.
• GRAC_SPM : TCode for Centralised FFighting -> You will see FFIDs assigned to you • /n/GRCPI/GRIA_EAM : TCode for DeCentralised FFighting -> You can see the FFIDs assigned to you	• GRAC_SPM : TCode for Centralised FFighting -> You will see FFROLEs assigned to you • /n/GRCPI/GRIA_EAM : TCode for DCentralised FFighting -> Not applicable so wont work

ID Based vs Role Based

Configuration in a nutshell

1. Create all EAM users or decide amongst the existing users who gets what EAM role using ‘SU01’
2. Create/customize all EAM roles using ‘PFCG’
3. Assign those roles to their respective users using ‘SU01’
4. Create an FFID/FFRole with the predetermined roles/tcodes using ‘SU01/PFCG/BRM’
5. Maintain GRC Plug-In System Configuration Parameters:
   - SPRO -> IMG -> GRC (Plug-In) -> Maintain Plug-In Configuration Settings
   -

Parameter ID	Parameter Value	Description
1000	Plug-in Connector ID	This information is used to connect to the Plug-In system.
4000	ID Based:1 Role Based:2	Application type
4001	Days	Default Firefighter Validity Period (Days)
4008	Yes/No	Send Firefighter Id Login Notification
4010	Z_SAP_GRAC_SPM_FF ID	Firefighter ID role name

6. Maintain GRC System Configuration Parameters:
   - SPRO -> IMG -> GRC -> AC-> Maintain Configuration Settings

Parameter ID	Parameter Value	Description
4000	ID Based:1 Role Based:2	Application type
4001	Days	Default Firefighter Validity Period (Days)
4002	Yes/No	Send Email Immediately
4003	Yes/No	Retrieve Change Log
4004	Yes/No	Retrieve System log
4005	Yes/No	Retrieve Audit log
4006	Yes/No	Retrieve OS Command log

ID Based vs Role Based		
Parameter ID	**Parameter Value**	**Description**
4007	Yes/No	Send Log Report Execution Notification Immediately
4008	Yes/No	Send Firefighter Id Login Notification
4009	Yes/No	Log Report Execution Notification
4010	Z_SAP_GRAC_SPM_FFID	Firefighter ID role name
4012	All Users:1 Controllers:2	Default users for forwarding the Audit Log workflow
4013	Yes/No	Firefighter ID owner can submit request for FF ID owned
4014	Yes/No	Firefighter ID controller can submit request for FF ID controlled
4015	Yes/No	Enable Decentralized Firefighting

7. Maintain User Exits
   - SPRO -> IMG -> GRC (Plug-In) -> Maintain User Exits
8. Maintain Connection Settings: SUPMG Integration scenario
   - SPRO -> IMG -> GRC -> Common Component Settings -> Integration Framework -> Maintain Integration Scenario
9. Activate/Check Criticality Level BC Set
   - SPRO -> IMG -> SCPR20 -> GRAC_SPM_CRITICALITY_LEVEL
10. Maintain Criticality level
    - SPRO -> IMG -> GRC -> AC-> EAM -> Maintain Criticality Levels for EAM
11. Run Synchronization jobs
    - SPRO -> IMG -> GRC -> AC-> Synchronization Jobs
      - Check for the help option to see what does what.
12. Schedule Background Jobs for EAM log collection on periodic basis
    - SM36 -> GRAC_SPM_LOG_SYNC_UPDATE
13. Maintain login/log notifications – only if you want to customize the default ones.
    - SPRO -> IMG -> GRC (Plug-In) -> Maintain Custom Notification/Text Messages for EAM (Plug-In)
14. Verify Time Zones of the Operating System and the AC server match to ensure EAM logs are

ID Based vs Role Based
captured ▪ SPRO -> IMG -> GRC -> General Settings -> Time Zones -> Maintain System Settings 15. Create/Maintain AC Owners ▪ NWBC -> Setup -> Access Owners -> Access Control Owners 16. Assign FFID/FFRoles to FF Owners ▪ NWBC -> Setup -> Superuser Assignment -> Owners 17. Assign FFID/FFRoles to end users (firefighter) and controllers ▪ NWBC -> Setup -> Superuser Assignment -> Firefighter IDs 18. Create Reason Codes ▪ NWBC -> Setup -> Superuser Maintenance -> Reason Codes Once all of the afore mentioned tasks are performed and successful, firefighter can perform firefighting tasks. His/her activities will be logged, which can be monitored by the Controller and viewed by relevant personnel. **** You might encounter problems in regards to FFID not showing up, Logs not getting collected properly etc. Please check the links provided for additional information.***

This pretty much is the gist of EAM. For a more comprehensive understanding/configuration and other bits and pieces on this topic, please check out the links in the following document put together by ***Alessandro***, which covers everything in detail. Please check under Emergency Access Management (EAM).

http://scn.sap.com/docs/DOC-57438

A big 'Thank You' to the people who created and made these posts available for the benefit of people like myself. Your time/effort is very much appreciated guys.

Regards,

Leo..